

Please type a plus sign (+) inside this box → ☐

PTO/SB/05 (12/97)
Approved for use through 09/30/00. OMB 0651-0032
Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

**UTILITY
PATENT APPLICATION
TRANSMITTAL**

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No. 042390.P5965

Total Pages 38

First Named Inventor or Application Identifier

Lance Hacking

Express Mail Label No.

EL105935317US

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents

ADDRESS TO:

Assistant Commissioner for Patents
Box Patent Application
Washington, DC 20231

1. ☒ Fee Transmittal Form
(Submit an original, and a duplicate for fee processing)
2. ☒ Specification [Total Pages 26]
(preferred arrangement set forth below)
- Descriptive title of the Invention
 - Cross References to Related Applications
 - Statement Regarding Fed sponsored R & D
 - Reference to Microfiche Appendix
 - Background of the Invention
 - Brief Summary of the Invention
 - Brief Description of the Drawings (if filed)
 - Detailed Description
 - Claim(s)
 - Abstract of the Disclosure
3. ☒ Drawing(s) (35 CFR 113) [Total Sheets 6]
4. Oath or Declaration [Total Pages]
- a. ☒ Newly executed (original copy)
- b. ☐ Copy from a prior application (37 CFR 1.63(d))
(for continuation/divisional with Box 17 completed)
[Note Box 5 below]
- i. ☐ **DELETION OF INVENTOR(S)**
Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b).
5. ☐ Incorporation By Reference (usable if Box 4b is checked)
The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under Box 4b, is considered as being part of the disclosure of the accompanying application and is hereby incorporated by reference therein.

6. ☐ Microfiche Computer Program (Appendix)
7. Nucleotide and/or Amino Acid Sequence Submission (if applicable, all necessary)
- a. ☐ Computer Readable Copy
- b. ☐ Paper Copy (identical to computer copy)
- c. ☐ Statement verifying identity of above copies

ACCOMPANYING APPLICATION PARTS

8. ☒ Assignment Papers (cover sheet & document(s))
9. ☐ 37 CFR 3.73(b) Statement (when there is an assignee) ☐ Power of Attorney
10. ☐ English Translation Document (if applicable)
11. ☐ Information Disclosure Statement (IDS)/PTO - 1449 ☐ Copies of IDS Citations
12. ☐ Preliminary Amendment
13. ☒ Return Receipt Postcard (MPEP 503)
(Should be specifically itemized)
14. ☐ Small Entity Statement filed in prior application, Statement(s) ☐ Status still proper and desired
15. ☐ Certified Copy of Priority Document(s)
(if foreign priority is claimed)
16. ☐ Other:

17. If a **CONTINUING APPLICATION**, check appropriate box and supply the requisite information:

☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No: _____/_____

18. CORRESPONDENCE ADDRESS

☐ Customer Number of Bar Code Label

(Insert Customer No. or Attach bare code label here)

or ☒ Correspondence address below

NAME

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP

ADDRESS

12400 Wilshire Boulevard, Seventh Floor

CITY

Los Angeles

STATE

California

ZIP CODE

90025

COUNTRY

U.S.A.

TELEPHONE

(714) 557-3800

FAX

(714) 557-3347

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231.

FEE TRANSMITTAL

Note: Effective October 1, 1997.
Patent fees are subject to annual revision.

TOTAL AMOUNT OF PAYMENT (\$) \$1,614.00

Complete if Known

Application Number	
Filing Date	
First Named Inventor	Lance Hacking, et al.
Group Art Unit	
Examiner Name	
Attorney Docket Number	042390.P5965

METHOD OF PAYMENT (check one)

1. ☒ The Commissioner is hereby authorized to charge indicated fees and credit any over payments to:

Deposit Account Number 02-2666
Deposit Account Name Blakely, Sokoloff, Taylor & Zafman LLP

- ☒ Charge Any Additional Fee Required Under 37 CFR 1.16 and 1.17 ☐ Charge the Issue Fee Set in 37 CFR 1.18 at the Mailing of the Notice of Allowance, 37 CFR 1.311(b)

2. ☒ Payment Enclosed:
☒ Check ☐ Money Order ☐ Other

FEE CALCULATION (fees effective 10/01/96)

1. FILING FEE

Large Entity Code	Large Entity Fee (\$)	Small Entity Code	Small Entity Fee (\$)	Fee Description	Fee Paid
101	790	201	395	Utility filing fee	\$790
106	330	206	165	Design filing fee	
107	540	207	270	Plant filing fee	
108	790	208	395	Reissue filing fee	
114	150	214	75	Provisional filing fee	

SUBTOTAL (1) (\$) 790.00

2. CLAIMS

Total Claims	Extra	Fee from below	Fee Paid
37	-20 = 17	X \$22.00 =	374.00
Independent Claims	8 -3 = 5	X \$82.00 =	410.00
Multiple Dependent Claims			

Large Entity Code	Large Entity Fee (\$)	Small Entity Code	Small Entity Fee (\$)	Fee Description
103	22	203	11	Claims in excess of 20
102	82	202	41	Independent claims in excess of 3
104	270	204	135	Multiple Dependent claim
109	82	209	41	Reissue independent claims over original patent
110	22	210	11	Reissue claims in excess of 20 and over original patent

SUBTOTAL (2) (\$) 784.00

FEE CALCULATION (continued)

3. ADDITIONAL FEE

Large Entity Code	Large Entity Fee (\$)	Small Entity Code	Small Entity Fee (\$)	Fee Description	Fee Paid
105	130	205	65	Surcharge - late filing fee or oath	
127	50	227	25	Surcharge - late provisional filing fee or cover sheet.	
139	130	139	130	Non-English specification	
147	2,520	147	2,520	For filing a request for reexamination	
112	920	112	920	Requesting publication of SIR prior to Examiner action	
113	1,840	113	1,840	Requesting publication of SIR after Examiner action	
115	110	215	55	Extension for response within first month	
116	400	216	200	Extension for response within second month	
117	950	217	475	Extension for response within third month	
118	1,510	218	755	Extension for response within fourth month	
119	310	219	155	Notice of Appeal	
120	310	220	155	Filing a brief in support of an appeal	
121	270	221	135	Request for oral hearing	
138	1,510	138	1,510	Petition to institute a public use proceeding	
140	110	240	55	Petition to revive unavoidably abandoned application	
141	1,320	241	660	Petition to revive unintentionally abandoned application	
142	1,320	242	660	Utility issue fee (or reissue)	
143	450	243	225	Design issue fee	
144	670	244	335	Plant issue fee	
122	130	122	130	Petitions to the Commissioner	
123	50	123	50	Petitions related to provisional applications	
126	240	126	240	Submission of Information Disclosure Stmt	
581	40	581	40	Recording each patent assignment per property (times number of properties)	\$40.00
146	790	246	395	Filing a submission after final rejection (37 CFR 1.129(a))	
149	790	249	395	For each additional invention to be examined (37 CFR 1.129(b))	

Other fee (specify) _____

Other fee (specify) _____

SUBTOTAL (2) (\$) 40.00

* Reduced by Basic Filing Fee Paid

SUBMITTED BY

Typed or Printed Name Kimberley G. Nobles

Signature Kimberley G. Nobles

Date 7/24/98

Complete (if applicable)

Reg. Number 38,255

Deposit Account User ID 02-2666

Burden Hour Statement: This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231.

UNITED STATES PATENT APPLICATION
FOR
A METHOD AND APPARATUS FOR PERFORMING
CACHE SEGMENT FLUSH AND
CACHE SEGMENT INVALIDATION OPERATIONS

Inventors: Lance Hacking
Shreekant S. Thakkar
Thomas Huff
Vladimir Pentkovski
Hsien-Cheng E. Hsieh

Prepared By:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN
12400 Wilshire Blvd., 7th Floor
Los Angeles, California 90025-1026
(714) 557-3800

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates in general to the field of computer systems, and in particular, to an apparatus and method for providing instructions which facilitate the invalidation and/or flushing of a portion of a cache memory within a cache system.

2. Description of the Related Art

The use of a cache memory with a computer system facilitates the reduction of memory access time. The fundamental idea of cache organization is that by keeping the most frequently accessed instructions and data in the fast cache memory, the average memory access time will approach the access time of the cache. To achieve the optimal tradeoffs between cache size and performance, typical computer systems implement a cache hierarchy, that is, different levels of cache memory. The different levels of cache correspond to different distances from the computer system core. The closer the cache is to the computer system, the faster the data access. However, the closer the cache is to the computer system, the more costly it is to implement. As a result, the closer the cache level, the faster and smaller the cache.

A cache unit is typically located between the computer system and main memory; it typically includes a cache controller and a cache memory such as a static random access memory (SRAM). The cache unit can be included on the same chip as the computer system or can exist as a separate component. Alternatively, the cache controller may be included on the computer system chip and the cache memory is formed by external SRAM chips.

The performance of cache memory is frequently measured in terms of its hit ratio. When the computer system refers to memory and finds the data in its cache, it is said to produce a hit. If the data is not found in cache, then it is in main memory and is counted as a miss. If a miss occurs, then an allocation is made at the entry indexed by the address of the access. The access can be for loading data to the computer system or storing data from the computer system to memory. The cached information is retained by the cache memory until it is no longer needed, made invalid or replaced by other data, in which instances the cache entry is de-allocated.

If other computer systems or system components have access to the main memory, as is the case, for example, with a DMA controller, and the main memory can be overwritten, the cache controller must inform the applicable cache that the data stored within the cache is invalid if the data in the main memory changes. Such an operation is known as cache invalidation. If the cache controller implements a write-back strategy and, with a cache hit, only writes data from the computer system to its cache, the cache content must be transferred to the main memory under specific conditions. This applies, for example, when the DMA chip transfers data from the main memory to a peripheral unit, but the current values are only stored in an SRAM cache. This type of operation is known as a cache flush.

Currently, such invalidating and/or flushing operations are performed automatically by hardware, for an associated cache line. In certain situations, software have been developed to invalidate and/or flush the cache memory. Currently, such software techniques involve the use of an instruction which operates on the entire cache memory corresponding to the computer system from which the instruction originated. However, such invalidation and/or flushing operations require a large amount of time to complete, and provides no granularity

or control for the user to invalidate and/or flush specific data or portions of data from the cache, while retaining the other data within the cache memory intact.

When a flushing operation operates only on the entire cache memory, it results in inflexibility and impacts system performance. In addition, where a cache

5 invalidation operation operates only on the entire cache, data corruption may result.

BRIEF SUMMARY OF THE INVENTION

A method and apparatus for including in a computer system, instructions for performing cache memory invalidate and cache memory flush operations. In one embodiment, the computer system comprises a cache memory having a plurality of cache lines each of which stores data, and a storage area to store a data operand. An execution unit is coupled to the storage area, and operates on data elements in the data operand to invalidate data in a predetermined portion of the plurality of cache lines in response to receiving a single instruction.

BRIEF DESCRIPTION OF THE DRAWINGS

The invention is illustrated by way of example, and not limitation, in the figures. Like reference indicate similar elements.

Figure 1 illustrates an exemplary computer system in accordance with one
5 embodiment of the invention.

Figure 2 illustrates one embodiment of the format of a cache control instruction 160 provided according to one embodiment of the invention.

Figure 3 illustrates the general operation of the cache control technique according to one embodiment of the invention.

10 Figure 4A illustrates one embodiment of the operation of the cache segment invalidate instruction 162.

Figure 4B illustrates one embodiment of the operation of the cache segment flush instruction 164.

15 Figure 5A is a flowchart illustrating one embodiment of the cache segment invalidate process of the present invention.

Figure 5B is a flowchart illustrating one embodiment of the cache segment flush process of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

In the following description, numerous specific details are set forth to provide a thorough understanding of the invention. However, it is understood that the invention may be practiced without these specific details. In other instances, well-known circuits, structures and techniques have not been shown in detail in order not to obscure the invention.

Figure 1 illustrates one embodiment of a computer system 100 which implements the principles of the present invention. Computer system 100 comprises a computer system 105, a storage device 110, and a bus 115. The computer system 105 is coupled to the storage device 110 by the bus 115. The storage device 110 represents one or more mechanisms for storing data. For example, the storage device 110 may include read only memory (ROM), random access memory (RAM), magnetic disk storage mediums, optical storage mediums, flash memory devices and/or other machine readable mediums. In addition, a number of user input/output devices, such as a keyboard 120 and a display 125, are also coupled to the bus 115. The computer system 105 represents a central processing unit of any type of architecture, such as CISC, RISC, VLIW, or hybrid architecture. In addition, the computer system 105 could be implemented on one or more chips. The storage device 110 represents one or more mechanisms for storing data. For example, the storage device 110 may include read only memory (ROM), random access memory (RAM), magnetic disk storage mediums, optical storage mediums, flash memory devices, and/or other machine-readable mediums. The bus 115 represents one or more buses (e.g., AGP, PCI, ISA, X-Bus, VESA, etc.) and bridges (also termed as bus controllers). While this embodiment is described in relation to a single computer

system computer system, the invention could be implemented in a multi-computer system computer system.

In addition to other devices, one or more of a network 130, a TV broadcast signal receiver 131, a fax/modem 132, a digitizing unit 133, a sound unit 134, and a graphics unit 135 may optionally be coupled to bus 115. The network 130 and fax modem 132 represent one or more network connections for transmitting data over a machine readable media (e.g., carrier waves). The digitizing unit 133 represents one or more devices for digitizing images (i.e., a scanner, camera, etc.). The sound unit 134 represents one or more devices for inputting and/or outputting sound (e.g., microphones, speakers, magnetic main memories, etc.). The graphics unit 135 represents one or more devices for generating 3-D images (e.g., graphics card). Figure 1 also illustrates that the storage device 110 has stored therein data 136 and software 137. Data 136 represents data stored in one or more of the formats described herein. Software 137 represents the necessary code for performing any and/or all of the techniques described with reference to Figures 2, and 4-6. Of course, the storage device 110 preferably contains additional software (not shown), which is not necessary to understanding the invention.

Figure 1 additionally illustrates that the computer system 105 includes decode unit 140, a set of registers 141, and execution unit 142, and an internal bus 143 for executing instructions. The computer system 105 further includes two internal cache memories, a level 0 (L0) cache memory which is coupled to the execution unit 142, and a level 1 (L1) cache memory, which is coupled to the L0 cache. An external cache memory, i.e., a level 2 (L2) cache memory 172, is coupled to bus 115 via a cache controller 170. The actual placement of the various cache memories is a design choice or may be dictated by the computer system architecture. Thus, it is

appreciated that the L1 cache could be placed external to the computer system 105. In alternate embodiments, more or less levels of cache (other than L1 and L2) may be implemented. It is appreciated that three levels of cache hierarchy are shown in Figure 1, but there could be more or less cache levels. For example, the present invention could be practiced where there is only one cache level (L0 only) or where there are only two cache levels (L0 and L1), or where there are four or more cache levels.

Of course, the computer system 105 contains additional circuitry, which is not necessary to understanding the invention. The decode unit 140, registers 141 and execution unit 142 are coupled together by internal bus 143. The decode unit 140 is used for decoding instructions received by computer system 105 into control signals and/or micro code entry points. In response to these control signals and/or micro code entry points, the execution unit 142 performs the appropriate operations. The decode unit 140 may be implemented using any number of different mechanisms (e.g., a look-up table, a hardware implementation, a PLA, etc.). While the decoding of the various instructions is represented herein by a series of if/then statements, it is understood that the execution of an instruction does not require a serial processing of these if/then statements. Rather, any mechanism for logically performing this if/then processing is considered to be within the scope of the implementation of the invention.

The decode unit 140 is shown including a fetching unit 150 which fetches instructions, and an instruction set 165 for performing operations on data. In one embodiment, the instruction set 165 includes a cache control instruction(s) provided in accordance with the present invention. In one embodiment, the cache control instructions include: a cache segment invalidate instruction(s) 162, a cache segment

flush instruction(s) 164 and a cache segment flush and invalidate instruction(s) 166 provided in accordance with the present invention. An example of the cache segment invalidate instruction(s) 162 includes a Page Invalidate (PGINVD) instruction which operates on a user specified linear address and invalidates the 4k
5 Byte physical page corresponding to the linear address from all levels of the cache hierarchy for all agents in the computer system that are connected to the computer system. An example of the cache segment flush instruction 164 includes a Page Flush (PGFLUSH) instruction 164 that flushes data in the 4 Kbyte physical page corresponding to the linear address on which the operation is performed. An
10 example of the cache segment flush and invalidate instruction 166 includes a Page Flush/Invalidate (PGFLUSHINV) instruction 166 that first flushes data in the 4 Kbyte physical page corresponding to the linear address on which the operation is performed, and then invalidates the 4 kilobyte physical page corresponding to the linear address. In alternative embodiments, the cache control instruction(s) may
15 operate on either a user specified linear or physical address and perform the associated invalidate and/or flush operations in accordance with the principles of the invention.

In addition to the cache segment invalidate instruction(s) 162, the cache segment flush instruction(s) 164, and the cache segment flush and invalidate
20 instruction(s) 166, computer system 105 can include new instructions and/or instructions similar to or the same as those found in existing general purpose computer systems. For example, in one embodiment the computer system 105 supports an instruction set which is compatible with the Intel® Architecture instruction set used by existing computer systems, such as the Pentium®II computer
25 system. Alternative embodiments of the invention may contain more or less, as well as different instructions and still utilize the teachings of the invention.

The registers 141 represent a storage are on computer system 105 for storing information, such as control/status information, scalar and/or packed integer data, floating point data, etc. It is understood that one aspect of the invention is the described instruction set. According to this aspect of the invention, the storage area used for storing the data is not critical. The term data processing system is used herein to refer to any machine for processing data, including the computer systems(s) described with reference to Figure 1.

Figure 2 illustrates one embodiment of the format of any one of the cache segment invalidate instruction 162, the cache segment flush instructions 164, and the cache segment flush and invalidate instruction 166 provided in accordance with the present invention. For discussion purposes, the instructions 162, 164 and 166 will be referred to as the cache control instruction 160. The cache control instruction 160 comprises an operational code (OP CODE) 210 which identifies the operation of the cache control instruction 160 and an operand 212 which specifies the name of a register or memory location which holds a starting address of the data object that the instruction 160 will be operating on.

Figure 3 illustrates the general operation of the cache control instruction 160 according to one embodiment of the invention. In the practice of the invention, the cache control instruction 160 provides the register (or memory) location which holds a starting address of the data object that the instruction 160 will be operating on. In one embodiment, the starting address includes X most significant bits, which are stored in the register (or memory) location, and Y least significant bits. The cache control process associated with the cache control instruction 160 then shifts the X bits to the right by Y bit positions to obtain the complete starting address. The cache control instruction 160 then operates on the data corresponding to the starting

address, and data corresponding to the Z subsequent addresses, in cache memory. In one embodiment, the cache control instruction 160 operates on one page of data stored in cache, of which the beginning address is stored in a register (or memory) location specified in the operand 212 of the cache control instruction. In alternate
5 embodiments, the cache control instruction 160 may operate on any predetermined amount of data stored in cache, of which the beginning address is stored in a register (or memory) location specified in the operand 212 of the cache control instruction.

In Figure 1, only L0, L1 and L2 levels are shown, but it is appreciated that more or less levels can be readily implemented. The embodiment shown in Figures
10 4-6 describes the use of the invention with respect to one cache level.

Details of various embodiments of the cache control instruction 160 will now be described. The cache segment invalidate instruction 162 will first be described. Figure 4A illustrates one embodiment of the cache segment invalidate instruction 162. Upon receiving the cache segment invalidate instruction 162, the computer
15 system 105 determines, from the operand 312 of the instruction 162, the register location in which the most significant bits of the starting address of the data object is stored. The computer system 105 then shifts the value in the operand 312, by the number of least significant bits of the starting address. Once the complete starting address is obtained, the computer system 105 sets the invalidate bit of the cache
20 memory 200 corresponding to the affected locations of the cache memory. In one embodiment, one page of the cache memory 220 having a starting address corresponding to that stored in the operand 312 will be invalidated. In alternate embodiments, data in any predetermined portions of the cache memory 220 having a starting address corresponding to that stored in the operand 312 will be invalidated
25 using the present technique.

The cache segment flush instruction 164 will next be described. Figure 4B illustrates one embodiment of the cache segment flush instruction 164. Upon receiving the cache segment flush instruction 164, the computer system 105 determines, from the operand 312 of the instruction 164, the register location in which the most significant bits of the starting address of the data object is stored. The computer system 105 then shifts the value in the operand 312, by the number of least significant bits of the starting address. Once the complete starting address is obtained, the computer system flushes the locations of cache memory 220 affected by execution of the instruction 164. In one embodiment, one page of the cache memory 220 having a starting address corresponding to that stored in the operand 312 will be flushed. In alternate embodiments, data in any predetermined portions of the cache memory 220 having a starting address corresponding to that stored in the operand 312 will be flushed.

The cache segment flush/invalidate instruction 166 will now be described. Figure 4C illustrates one embodiment of the cache segment flush and invalidate instruction 166. Upon receiving the cache segment flush instruction 166, the computer system 105 determines, from the operand 312 of the instruction 164, the register location in which the most significant bits of the starting address of the data object is stored. The computer system 105 then shifts the value in the operand 312, by the number of least significant bits of the starting address. Once the complete starting address is obtained, the computer system flushes the locations of cache memory 220 affected by execution of the instruction 164. In one embodiment, one page of the cache memory 220 having a starting address corresponding to that stored in the operand 312 will be flushed. In alternate embodiments, any predetermined portions of the cache memory 220 having a starting address corresponding to that stored in the operand 312 will be flushed. Next, the computer system 105

invalidates the affected areas of the cache memory 220 that have been flushed. In one embodiment, this is performed by setting the invalidate bit of each affected cache line.

Figure 5A is a flowchart illustrating one embodiment of the cache segment invalidate process of the present invention. Beginning from a start state, the process 500 proceeds to process block 510, where it examines the operand 312 of the instruction 62 received by the computer system 105 to determine the storage location of the value representing the most significant bits of the starting address of the corresponding operation. The process 500 then proceeds to process block 512, where it retrieves the value representing the most significant bits of the starting address from the storage location specified. The process 500 then advances to process block 514, where it shifts the retrieved value by a predetermined number of bits. In one embodiment, the predetermined number represents the number of least significant bits in the starting address. Next, the process 500 determines the cache segment affected by the operation or the instruction 162, as shown in process block 516. In one embodiment, the cache segment is a page. In one embodiment, a page contains 4k Bytes. In alternate embodiments, the cache segment may be any predetermined portion of the cache memory. The process 500 then proceeds to process block 516, where it invalidates the data in the corresponding cache segment beginning at the starting address specified. In one embodiment, this is performed by setting the invalid bit corresponding to each cache line in the cache segment. The process 500 then terminates.

Figure 5B is a flowchart illustrating one embodiment of the cache segment flush process of the present invention. Beginning from a start state, the process 520 proceeds to process block 522, where it examines the operand 312 of the instruction

64 or 66 received by the computer system 105 to determine the storage location of the value representing the most significant bits of the starting address of the corresponding operation. The process 520 then proceeds to process block 524, where it retrieves the value representing the most significant bits of the starting address from the storage location specified. The process 520 then advances to process block 526, where it shifts the retrieved value by a predetermined number of bits. In one embodiment, the predetermined number represents the number of least significant bits in the starting address. Next, the process 520 determines the cache segment affected by the operation or the instruction 64 or 66, as shown in process block 528.

10 In one embodiment, the cache segment is a page. In alternate embodiments the cache segment may be any predetermined portion of the cache. The process 520 then proceeds to process block 530, where it flushes the contents of the cache segment to the storage device specified. The process 520 then proceeds to decision block 530, where it queries if the instruction received corresponding to the operation is a FLUSH or a FLUSH and INVALIDATE instruction. If the instruction is a FLUSH, the process 520 terminates. If the instruction is a FLUSH and INVALIDATE instruction, the process 520 proceeds to process block 534, where it invalidates the data in the corresponding cache segment beginning at the starting address specified.

15 In one embodiment, this is performed by setting the invalid bit corresponding to each cache line in the cache segment. The process 520 then terminates.

20

The use of the present invention thus enhances system performance by providing an invalidate instruction and/or a flush instruction for invalidating and/or flushing data in any predetermined portion of the cache memory. For cases where consistency between the cache and main memory are maintained by software,

25 system performance is enhanced, since flushing only the affected portions of cache is more efficient and flexible than flushing the entire cache. In addition, system

performance is enhanced by having a flushing and/or invalidate operation that has a granularity that is larger than a cache line size, since the user can flush and/or invalidate a memory region using a single instruction instead of having to alter the code, as the computer system changes the size of a cache line.

5 While a preferred embodiment has been described, it is to understood that the invention is not limited to such use. In addition, while the invention has been described in terms of several embodiments, those skilled in the art will recognize that the invention is not limited to the embodiments described. The method and apparatus of the invention can be practiced with modification and alteration within
10 the spirit and scope of the appended claims. The description is thus to be regarded as illustrative instead of limiting on the invention.

CLAIMS

What is claimed is:

- 1 1. A computer system comprising:
2 a cache memory having a plurality of cache lines each of which stores
3 data;
4 a storage area to store a data operand; and
5 an execution unit coupled to said storage area to operate on data
6 elements in said data operand to invalidate data in a predetermined portion of the
7 plurality of cache lines in response to receiving a single instruction.
- 8 2. The computer system of Claim 1, wherein the data operand is a register
9 location.
- 10 3. The computer system of Claim 2, wherein the register location contains
11 a portion of a starting address of the cache line in which data is to be invalidated.
- 12 4. The computer system of Claim 3, wherein the portion of the starting
13 address includes a plurality of most significant bits of the starting address.

1 5. The computer system of Claim 4, wherein execution unit shifts the
2 data elements by a predetermined number of bit positions to obtain the starting
3 address of the cache line in which data is to be invalidated.

1 6. The computer system of Claim 1, wherein the predetermined portion
2 of the plurality of cache lines is a page in the cache memory.

1 7. A computer system comprising:
2 a first storage area to store data;
3 a cache memory having a plurality of cache lines each of which stores
4 data;
5 a second storage area to store a data operand; and
6 an execution unit coupled to said first storage area, said second storage
7 area, and said cache memory, said execution unit to operate on data elements in said
8 data operand to copy data from a predetermined portion of the plurality of cache
9 lines in the cache memory to the first storage area, in response to receiving a single
10 instruction.

1 8. The computer system of Claim 7, wherein the data operand is a register
2 location.

1 9. The computer system of Claim 8, wherein the register location contains
2 a plurality of most significant bits of a starting address of the cache line in which
3 data is to be copied.

1 10. The computer system of Claim 9, wherein execution unit shifts the
2 data elements by a predetermined number of bit positions to obtain the starting
3 address of the cache line in which data is to be copied.

1 11. The computer system of Claim 7, wherein the predetermined portion
2 of the plurality of cache lines is a page in the cache memory.

1 12. The computer system of Claim 7, wherein the execution unit further
2 invalidates data in the predetermined portion of the plurality of cache lines in
3 response to receiving the single instruction, upon copying the data to the first
4 storage area.

1 13 A processor comprising:
2 a decoder configured to decode instructions, and
3 a circuit coupled to said decoder, said circuit in response to a single
4 decoded instruction being configured to:

5 obtain a starting address of a predetermined area of a cache
6 memory on which the instruction will be performed;
7 invalidate data in the predetermined area of cache memory.

1 14. The processor of Claim 13, wherein a portion of the starting address is
2 located in a register specified in the decoded instruction.

1 15. The processor of Claim 13, wherein the portion of the starting address
2 includes a plurality of most significant bits of the starting address.

1 16. The processor of Claim 15, wherein the circuit shifts the data elements
2 by a predetermined number of bit positions to obtain the starting address of the
3 cache line in which data is to be invalidated.

1 17. The processor of Claim 13, wherein the predetermined portion of the
2 plurality of cache lines is a page in the cache memory.

1 18. A processor comprising:
2 a decoder configured to decode instructions, and
3 a circuit coupled to said decoder, said circuit in response to a single
4 decoded instruction being configured to:

5 obtain a starting address of a predetermined area of a cache
6 memory on which the instruction will be performed;
7 copy data in the predetermined area of cache memory;
8 store the copied data in a storage area separate from the cache memory.

1 19. The processor of Claim 18, wherein a portion of the starting address is
2 located in a register specified in the decoded instruction.

1 20. The processor of Claim 18, wherein the portion of the starting address
2 includes a plurality of most significant bits of the starting address.

1 21. The processor of Claim 20, wherein the circuit shifts the data elements
2 by a predetermined number of bit positions to obtain the starting address of the
3 cache line in which data is to be copied.

1 22. The processor of Claim 20, wherein the predetermined portion of the
2 plurality of cache lines is a page in the cache memory.

1 23. The processor of Claim 20, wherein said circuit further invalidates the
2 data in the predetermined portion of the plurality of cache lines in response to
3 receiving the single instruction, upon copying the data to the storage area.

1 24. A computer-implemented method, comprising:

2 a) decoding a single instruction;

3 b) in response to said step of decoding the single instruction,

4 obtaining a starting address of a predetermined area of a cache memory on which
5 the single instruction will be performed; and

6 c) completing execution of said single instruction by invalidating
7 data in the predetermined area of cache memory.

1 25. The method of Claim 24, wherein c) comprises setting an invalid bit
2 corresponding to the predetermined area of cache memory.

1 26. The method of Claim 24, wherein b) comprises:

2 b.1) obtaining a portion of the starting address from a storage
3 location specified in the decoded instruction;

4 b.2) shifting the portion of the starting address by a predetermined
5 number of bit positions to obtain the starting address of the cache line in which data
6 is to be invalidated.

1 27. The method of Claim 26, wherein in b.1) the portion of the starting
2 address contains a plurality of most significant bits of the starting address, and

3 wherein in b.2), the predetermined number of bit positions represent the number of
4 least significant bits of the starting address.

1 28. The method of Claim 24, wherein the predetermined portion of the
2 plurality of cache lines is a page in the cache memory.

1 29. A computer-implemented method, comprising:

2 a) decoding a single instruction;

3 b) in response to said step of decoding the single instruction,
4 obtaining a starting address of a predetermined area of a cache memory on which
5 the single instruction will be performed; and

6 c) completing execution of said single instruction by copying data
7 in the predetermined area of cache memory and storing the copied data in a storage
8 area separate from the cache memory.

1 30. The method of Claim 29, wherein c) comprises setting an invalid bit
2 corresponding to the predetermined area of cache memory.

1 31. The method of Claim 29, wherein b) comprises:

2 b.1) obtaining a portion of the starting address from a storage
3 location specified in the decoded instruction;

4 b.2) shifting the portion of the starting address by a predetermined
5 number of bit positions to obtain the starting address of the cache line in which data
6 is to be invalidated.

1 32. The method of Claim 31, wherein in b.1) the portion of the starting
2 address contains a plurality of most significant bits of the starting address, and
3 wherein in b.2), the predetermined number of bit positions represent the number of
4 least significant bits of the starting address.

1 33. The method of Claim 29, wherein the predetermined portion of the
2 plurality of cache lines is a page in the cache memory.

1 34. The method of Claim 29, further comprising:
2 d) invalidating the data in the predetermined portion of the
3 plurality of cache lines in response to receiving the single instruction, upon copying
4 the data to the storage area.

1 35. A computer-readable apparatus, comprising:
2 a computer-readable medium that stores an instruction which when executed
3 by a processor causes said processor to:
4 obtain a starting address of a predetermined area of a cache memory on
5 which the instruction will be performed; and

6 invalidate data in the predetermined area of cache memory.

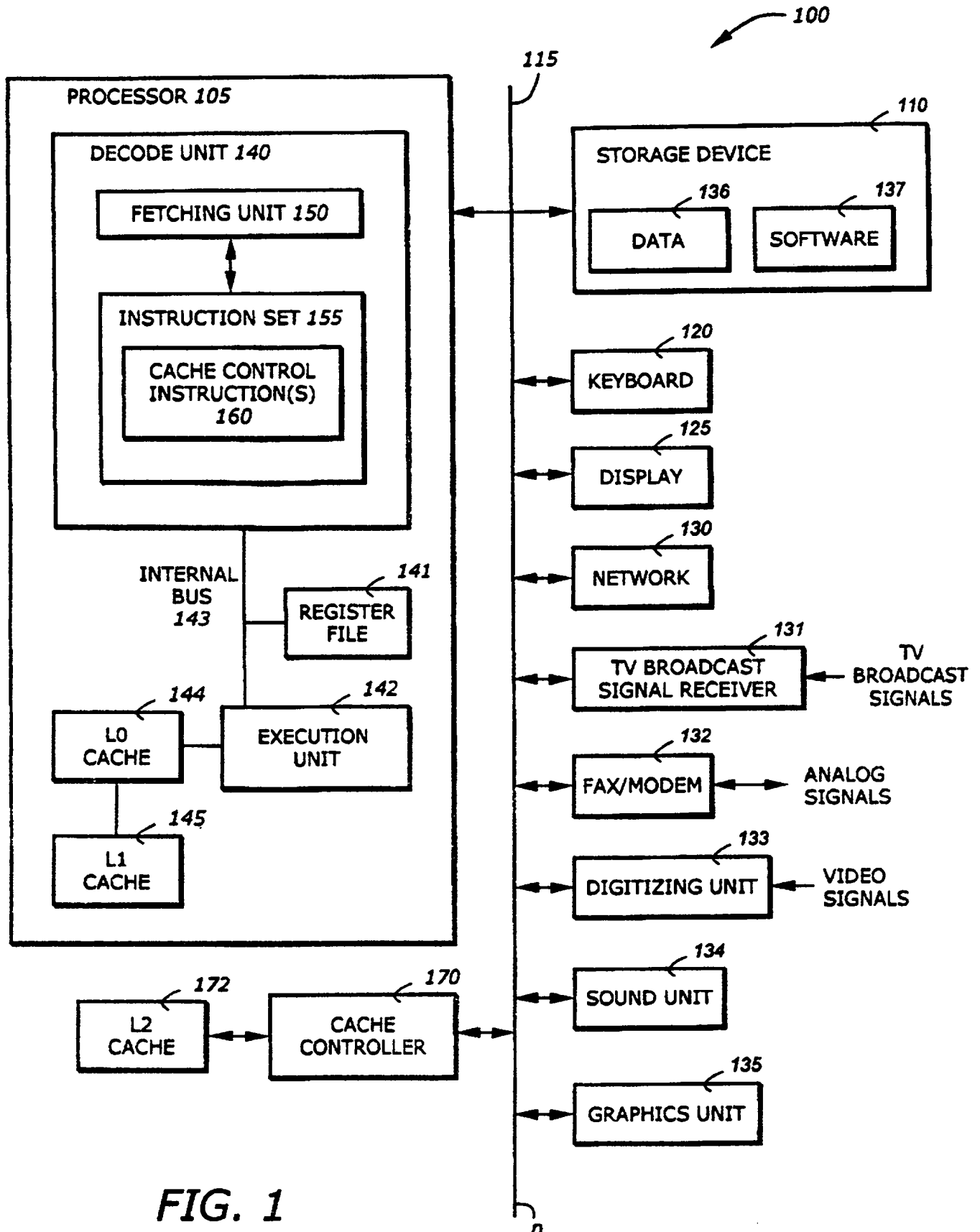
1 36. A computer-readable apparatus comprising:
2 a computer-readable medium that stores an instruction which when executed
3 by a processor causes said processor to:
4 obtain a starting address of a predetermined area of a cache memory on
5 which the instruction will be performed;
6 copy data from the predetermined area of cache memory; and
7 store the copied data in a storage area separate from the cache memory.

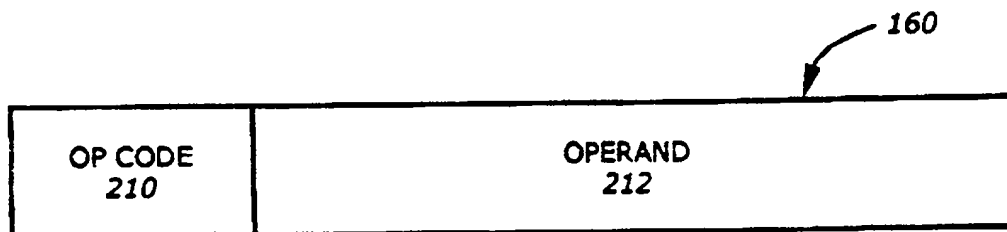
8
9
10
11
12 37. The apparatus of Claim 36, wherein the instruction further causes the
13 processor to:
14 invalidate the data in the predetermined portion of the plurality of cache
15 lines in response to receiving the instruction, upon copying the data to the storage
16 area.

ABSTRACT

A method and apparatus for including in a computer system, instructions for performing cache memory invalidate and cache memory flush operations. In one embodiment, the computer system comprises a cache memory having a plurality of cache lines each of which stores data, and a storage area to store a data operand. An execution unit is coupled to the storage area, and operates on data elements in the data operand to invalidate data in a predetermined portion of the plurality of cache lines in response to receiving a single instruction.

Patent Application No. 10/000,000



*FIG. 2*

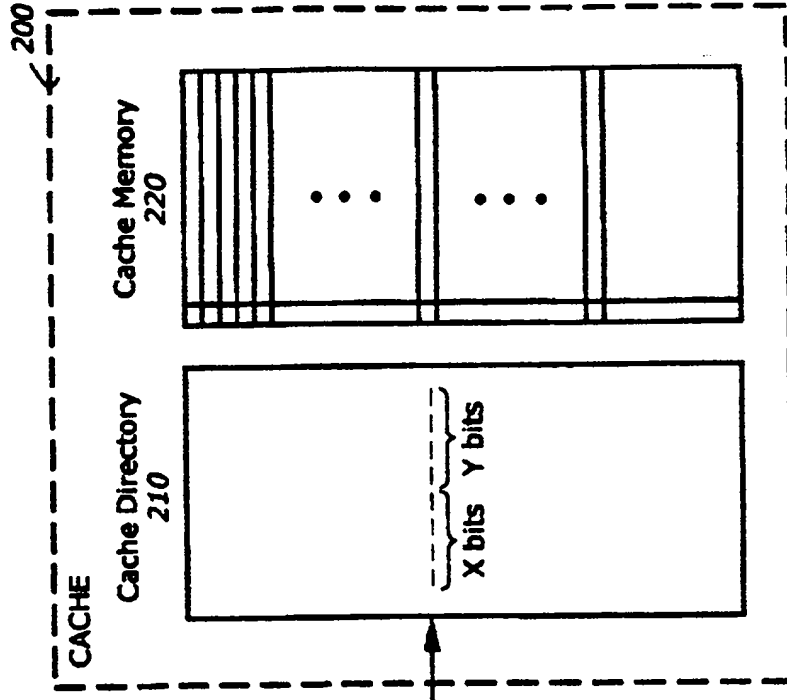
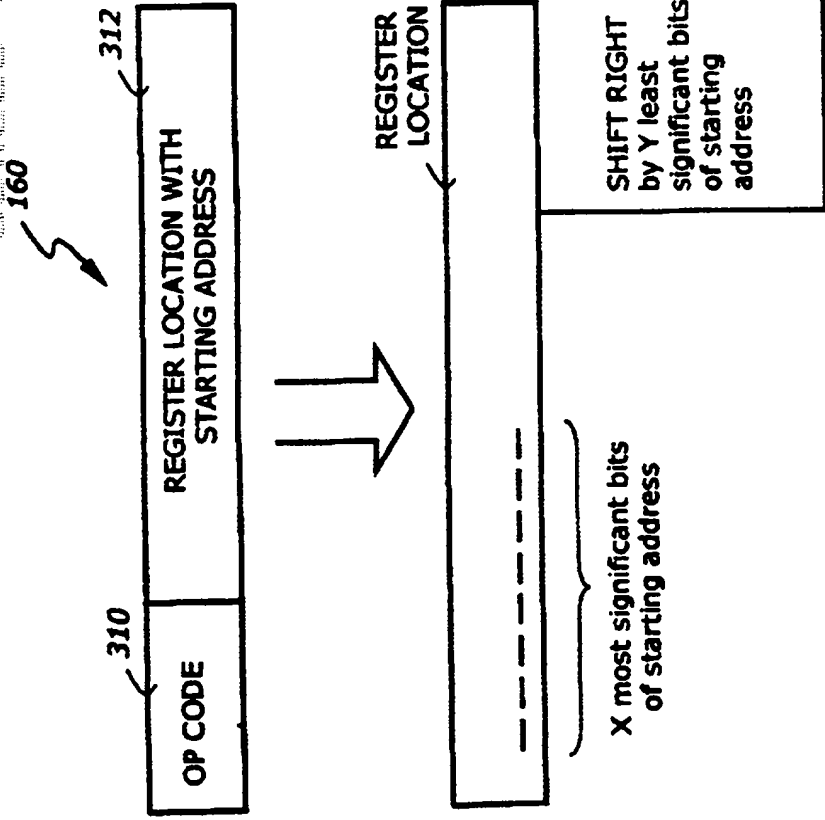
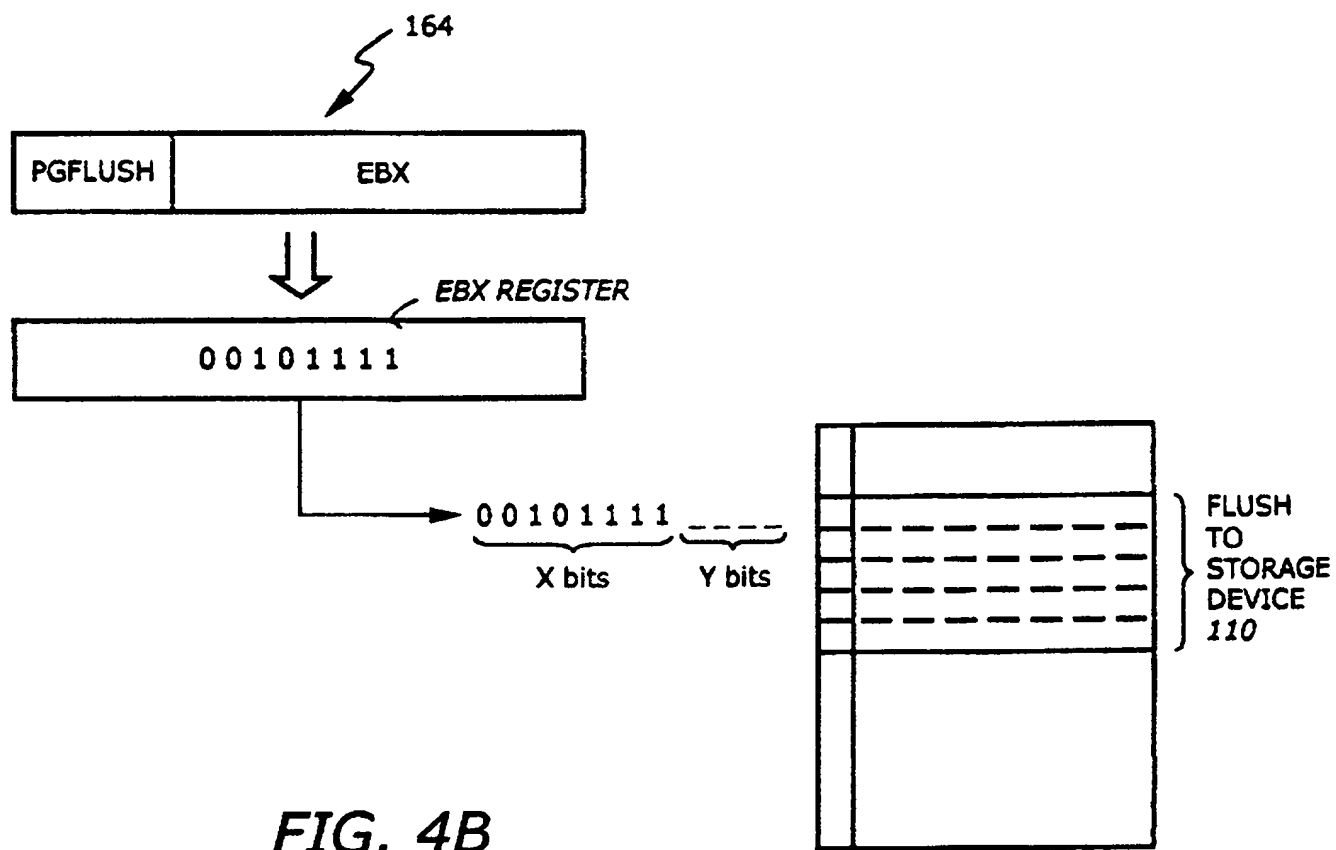
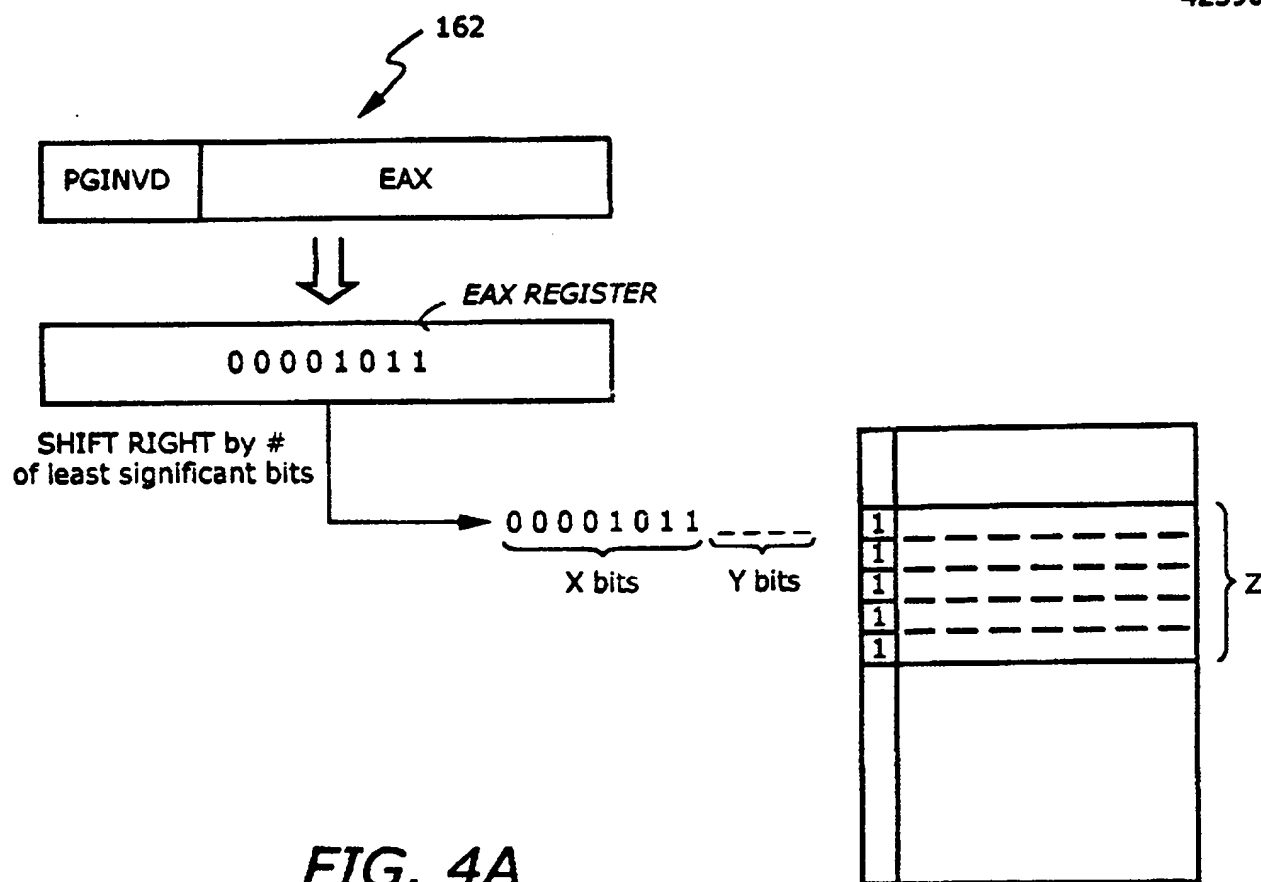


FIG. 3



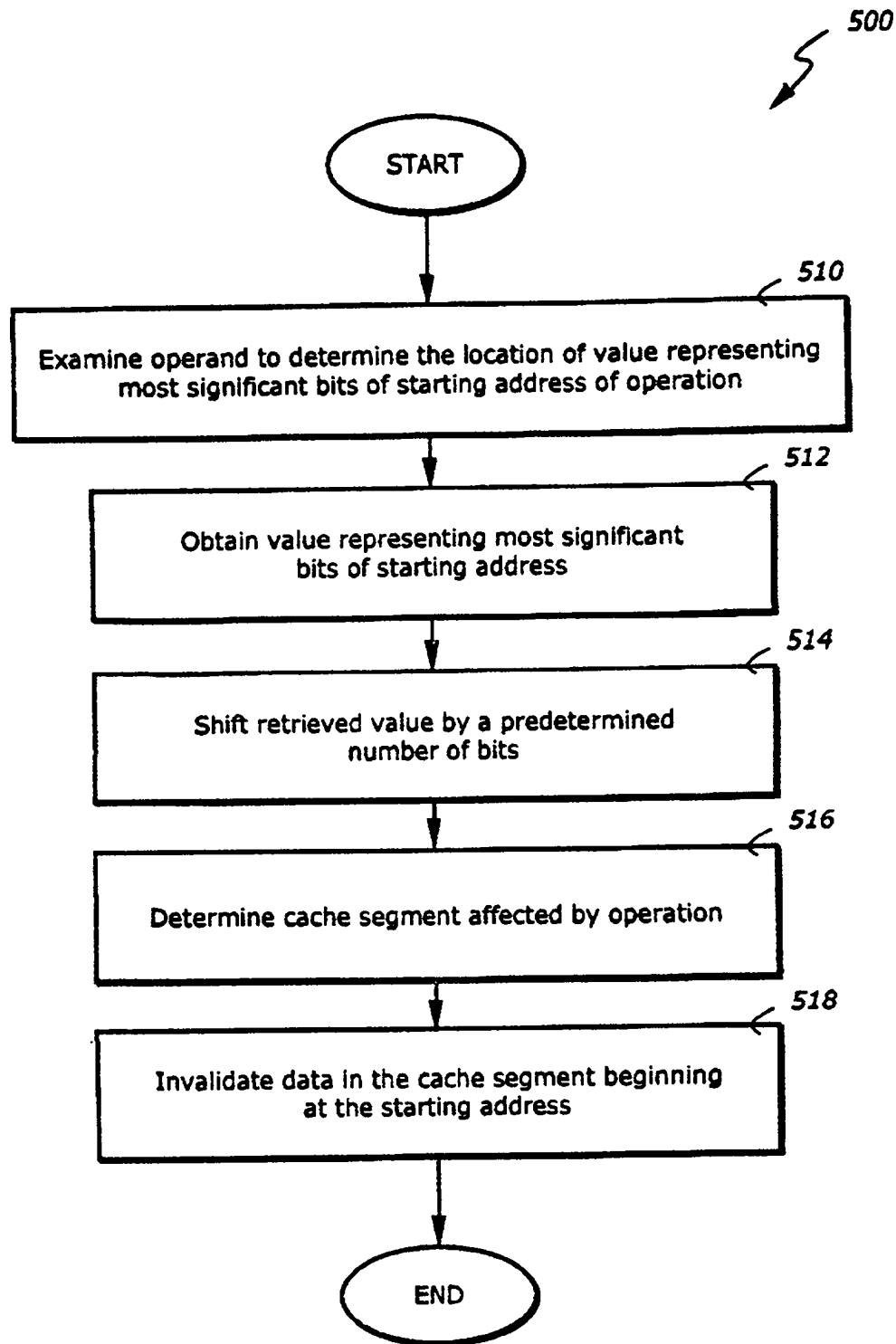


FIG. 5A

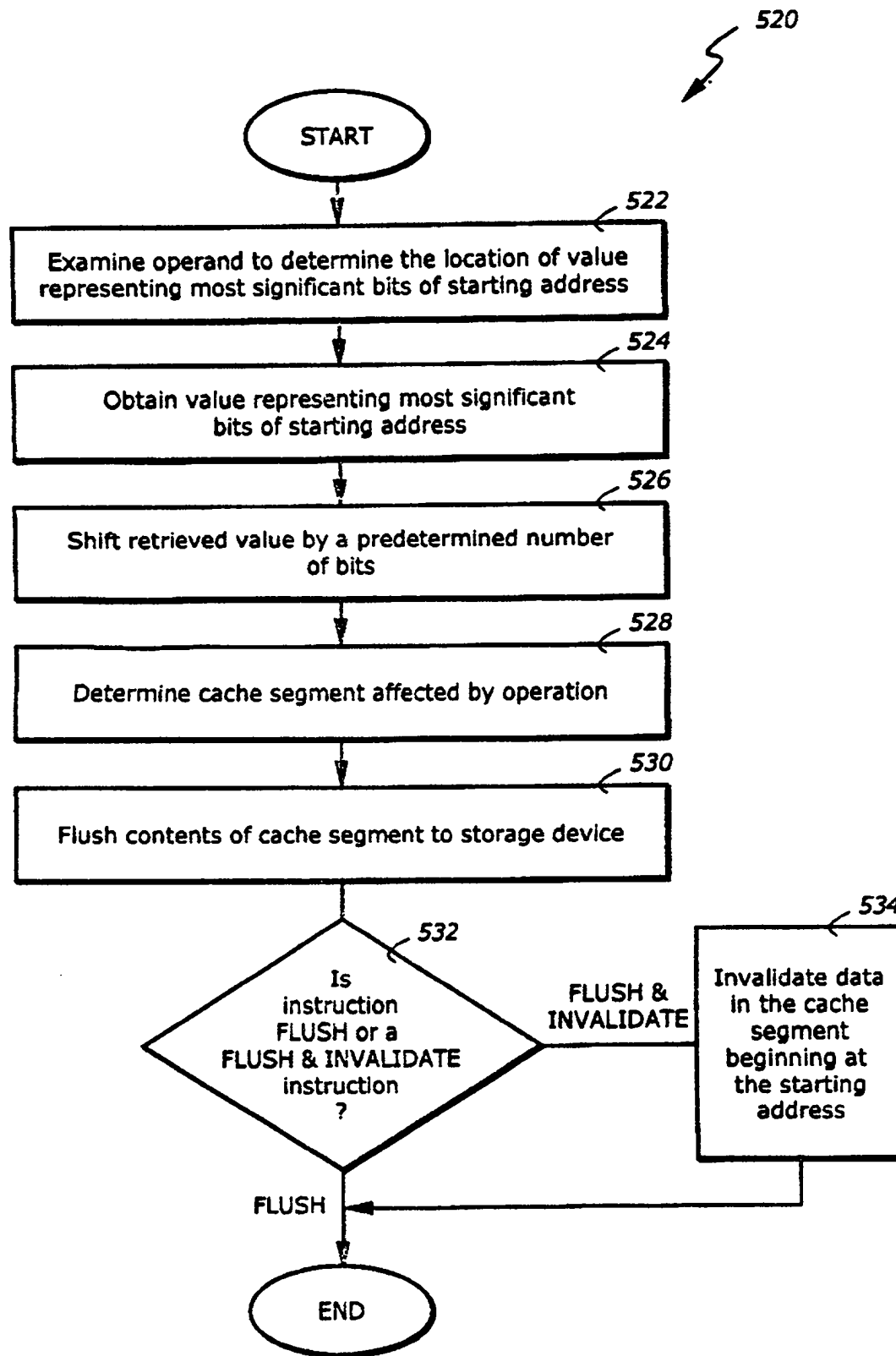


FIG. 5B

**DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION
(FOR INTEL CORPORATION PATENT APPLICATIONS)**

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below, next to my name.

I believe I am the original, first, and sole inventor (if only one name is listed below) or an original, first, and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

**A METHOD AND APPARATUS FOR PERFORMING CACHE SEGMENT FLUSH
AND CACHE SEGMENT INVALIDATION OPERATIONS**

the specification of which

☒ is attached hereto.
☐ was filed on _____ as
United States Application Number _____
or PCT International Application Number _____
and was amended on _____ .
(if applicable)

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claim(s), as amended by any amendment referred to above. I do not know and do not believe that the claimed invention was ever known or used in the United States of America before my invention thereof, or patented or described in any printed publication in any country before my invention thereof or more than one year prior to this application, that the same was not in public use or on sale in the United States of America more than one year prior to this application, and that the invention has not been patented or made the subject of an inventor's certificate issued before the date of this application in any country foreign to the United States of America on an application filed by me or my legal representatives or assigns more than twelve months (for a utility patent application) or six months (for a design patent application) prior to this application.

I acknowledge the duty to disclose all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119(a)-(d), of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s):

APPLICATION NUMBER	COUNTRY (OR INDICATE IF PCT)	DATE OF FILING (day, month, year)	PRIORITY CLAIMED UNDER 37 USC 119
			<input type="checkbox"/> No <input type="checkbox"/> Yes
			<input type="checkbox"/> No <input type="checkbox"/> Yes
			<input type="checkbox"/> No <input type="checkbox"/> Yes

I hereby claim the benefit under Title 35, United States Code, Section 119(e) of any United States provisional application(s) listed below:

APPLICATION NUMBER	FILING DATE

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, Section 112, I acknowledge the duty to disclose all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56 which became available between the filing date of the prior application and the national or PCT international filing date of this application:

APPLICATION NUMBER	FILING DATE	STATUS (ISSUED, PENDING, ABANDONED)

I hereby appoint BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP, a firm including: Farzad E. Amini, Reg. No. P42,261; Aloysius T. C. AuYeung, Reg. No. 35,432; William Thomas Babbitt, Reg. No. 39,591; Carol F. Barry, P41,600; Jordan Michael Becker, Reg. No. 39,602; Bradley J. Berezna, Reg. No. 33,474; Michael A. Bernadieu, Reg. No. 35,934; Roger W. Blakely, Jr., Reg. No. 25,831; Gregory D. Caldwell, Reg. No. 39,926; Kent M. Chen, Reg. No. 39,630; Lawrence M. Cho, Reg. No. 39,942; Yong S. Choi, Reg. No. P43,324; Thomas M. Coester, Reg. No. 39,637; Roland B. Cortes, Reg. No. 39,152; Barbara Bokanov Courtney, Reg. No. P42,442; William Donald Davis, Reg. No. 38,428; Michael Anthony DeSanctis, Reg. No. 39,957; Daniel M. De Vos, Reg. No. 37,813; Tarek N. Fahmi, Reg. No. P41,402; James Y. Go, Reg. No. 40,621; Richard Leon Gregory, Jr., P42,607; Dinu Gruia, Reg. No. P42,996; David R. Halvorson, Reg. No. 33,395; Thomas A. Hassing, Reg. No. 36,159; Phuong-Quan Hoang, P41,839; Willmore F. Holbrow III, Reg. No. P41,845; George W. Hoover II, Reg. No. 32,992; Eric S. Hyman, Reg. No. 30,139; Dag H. Johansen, Reg. No. 36,172; William W. Kidd, Reg. No. 31,772; Tim L. Kitchen, Reg. No. P41,900; Michael J. Mallie, Reg. No. 36,591; Paul A. Mendonsa P42,879; Darren J. Milliken, P42,004; Thinh V. Nguyen, Reg. No. P42,034; Kimberley G. Nobles, Reg. No. 38,255; Michael A. Proksch P43,021; Babak Redjaian, Reg. No. P42,096; James H. Salter, Reg. No. 35,668; William W. Schaal, Reg. No. 39,018; James C. Scheller, Reg. No. 31,195; Anand Sethuraman, Reg. No. P43,351; Charles E. Shemwell, Reg. No. 40,171; Maria McCormack Sobrino, Reg. No. 31,639; Stanley W. Sokoloff, Reg. No. 25,128; Allan T. Sponseller, Reg. No. 38,318; Steven R. Sponseller, Reg. No. 39,384; Geoffrey T. Staniford, P43,151; Judith A. Szepesi, Reg. No. 39,393; Vincent P. Tassinari, Reg. No. P42,179; Edwin H. Taylor, Reg. No. 25,129; George G. C. Tseng, Reg. No. 41,355; Lester J. Vincent, Reg. No. 31,460; John Patrick Ward, Reg. No. 40,216; Stephen Warhola, P43,237; Ben J. Yorks, Reg. No. 33,609; and Norman Zafman, Reg. No. 26,250; my attorneys; and Amy M. Armstrong, Reg. No. P42,265; Robert Andrew Diehl, Reg. No. P40,992; and Edwin A. Sloane, Reg. No. 34,728; my patent agents, of BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN, LLP with offices located at 12400 Wilshire Boulevard, 7th Floor, Los Angeles, California 90025, telephone (714) 557-3800, and Joseph R. Bond, Reg. No. 36,458; Richard C. Calderwood, Reg. No. 35,468; Sean Fitzgerald, Reg. No. 32,027; Naomi Obinata, Reg. No. 39,320; Thomas C. Reynolds, Reg. No. 32,488; Steven P. Skabrat, Reg.

Send correspondence to Kimberley G. Nobles, Reg. No. 38,255, BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP, 12400 Wilshire Boulevard, 7th Floor, Los Angeles, California 90025 and direct telephone calls to Kimberley G. Nobles, Reg. No. 38,255, (714) 557-3800.
(Name of Attorney or Agent)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Full Name of Sole/First Inventor (given name, family name) Lance Hacking


Inventor's Signature [Signature] Date 1 July 98

Residence Portland, Oregon Citizenship U.S.A.
(City, State) (Country)

P. O. Address 15739 N.W. Rondos Drive

Portland, Oregon 97229 U.S.A.

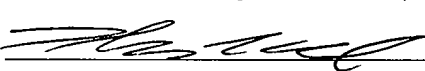
Full Name of Second/Joint Inventor (given name, family name) Shreekant Thakkar

Inventor's Signature  Date 7/10/98

Residence Portland, Oregon Citizenship British
(City, State) (Country)

P. O. Address 150 S.W. Moonridge Place
Portland, Oregon 97225 U.S.A.

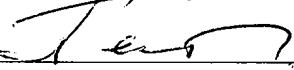
Full Name of Third/Joint Inventor (given name, family name) Thomas Huff

Inventor's Signature  Date 7/6/98

Residence Portland, Oregon Citizenship U.S.A.
(City, State) (Country)

P. O. Address 618 N.W. 12th Street, #310
Portland, Oregon 97209 U.S.A.

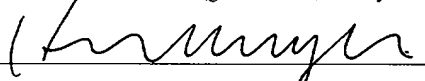
Full Name of Fourth/Joint Inventor (given name, family name) Vladimir Pentkovski

Inventor's Signature  Date July 16/98

Residence Folsom, California Citizenship Russia
(City, State) (Country)

P. O. Address 221 Luna Circle
Folsom, California 95630 U.S.A.

Full Name of Fifth/Joint Inventor (given name, family name) Hsien-Cheng E. Hsieh

Inventor's Signature  Date 07/15/98

Residence Gold River, California Citizenship Taiwan
(City, State) (Country)

P. O. Address 2078 Yellow Aster Court
Gold River, California 95670 U.S.A.